The Thesis Committee for David Antonio Vargas
certifies that this is the approved version of the following thesis:

Diagonal Plus Low Rank Approximation of Matrices for Solving Modal Frequency Response Problems

APPROVED BY

SUPERVISING COMMITTEE:

Jeffrey K. Bennighof, Supervisor

Jayant Sirohi

Diagonal Plus Low Rank Approximation of Matrices for Solving Modal Frequency Response Problems

by

David Antonio Vargas, B.S.As.E

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ENGINEERING

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2010

Dedicated to my parents, my brother and my grandmother.

Acknowledgments

I wish to thank my supervisor, Dr. Jeffrey K. Bennighof. His patience, encouragement and generous support have made possible the development of this thesis.

I would also like to thank Mark Muller, Jeremiah Palmer and Qinqin Li, whose knowledge was always shared with me.

Finally I would like to thank Jack Poulsen and Joshua Haben. Their support during the early months of graduate school was priceless.

Diagonal Plus Low Rank Approximation of Matrices for Solving Modal Frequency Response Problems

David Antonio Vargas, MSE
The University of Texas at Austin, 2010

Supervisor: Jeffrey K. Bennighof

If a structure is composed mainly of one material but contains a small amount of a second material, and if these two materials have significantly different levels of structural damping, this can increase the cost of solving the modal frequency response problem substantially. Even if the rank of the contribution to the finite element structural damping matrix from the second material is very low, the matrix becomes fully populated when transformed to the modal representation. As a result, the complex-valued modal matrix that represents the structure's stiffness and structural damping is both full rank, because of the diagonal part contributed by the stiffness, and fully populated, because of off-diagonal imaginary terms contributed by the second material's structural damping. Solving the modal frequency response problem at many frequencies requires either the factorization of a coefficient matrix at every frequency, or the solution of a complex symmetric eigenvalue problem associated with the modal stiffness/structural damping matrix. The cost of both of these

approaches is proportional to the cube of the number of modes included in the analysis. This cost could be reduced greatly if the damping properties of the structure were handled carefully in modeling the structure, but in practical computation of the modal frequency response, the information that could potentially reduce the computational cost is often unavailable.

This thesis explores the possibilities of obtaining a representation of the complex modal stiffness/structural damping matrix as a diagonal matrix plus a matrix of minimal rank. An algorithm for computing a "diagonal plus low rank" (DPLR) representation is developed, along with an iterative algorithm for using an inexact DPLR approximation in the solution of the modal frequency response problem. The behavior of these algorithms is investigated on several example problems.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

For a structure that is composed mainly of one material, but contains a small amount of one or more other materials whose structural damping behavior differs from that of the first material, finite element discretization results in the equations of motion

$$M\ddot{\mathbf{x}} + (1 + i\gamma)K\mathbf{x} + iK_s\mathbf{x} = \mathbf{F}. \tag{1.1}$$

where γ is the structural damping constant of the predominant material in the structure and K_s is a structural damping matrix representing how structural damping levels differ from γ for the other materials. The rank of K_s can be low since the non-predominant materials are not present in the entire structure but only in parts of it.

If the excitation and the response are harmonic in time, they can be written as $\mathbf{F} = \mathbf{F}e^{i\omega t}$ and $\mathbf{x} = \mathbf{x}e^{i\omega t}$, where ω is the excitation frequency. Then the equations of motion become those of the frequency response problem (FRP).

$$\left(-\omega^2 M + K(1 + i\gamma) + iK_s\right)\mathbf{x} = \mathbf{F}. \tag{1.2}$$

For frequency sweep analysis for which there are many excitation frequencies, solving the frequency response problem becomes expensive, particularly

if there are many degrees of freedom in the finite element discretization, because the coefficient matrix must be factored for each excitation frequency. The high cost of this approach is typically avoided by solving the eigenvalue problem

$$K\Phi = M\Phi\Lambda, \tag{1.3}$$

in which Φ is a rectangular matrix containing eigenvectors and Λ is a diagonal matrix containing eigenvalues corresponding to natural frequencies up to a specified cutoff frequency, typically chosen based on the highest excitation frequency of interest. The solution of the FRP is approximated in terms of the eigenvectors as $\mathbf{x} \approx \Phi\boldsymbol{\eta}$, and the frequency response equation of motion is pre-multiplied by Φ^T, yielding the modal FRP

$$\left[-\omega^2 I + \Lambda(1 + i\gamma) + i\Phi^T K_s \Phi\right] \boldsymbol{\eta} = \Phi^T \mathbf{F}, \tag{1.4}$$

which is solved for the modal displacements $\boldsymbol{\eta}$. The solution of the FRP is finally obtained by multiplying $\boldsymbol{\eta}$ by Φ.

The solution of the modal FRP is very inexpensive if all of the matrices on the left hand side of the equation are diagonal. The structural damping matrix γK is diagonalized by the eigenvectors, but $\Phi^T K_s \Phi$ is typically fully populated, even though its rank cannot be greater than that of K_s. If the fully populated coefficient matrix is handled by factoring it at every frequency, the computational cost at each frequency is proportional to the cube of the number of modes included in the analysis. Alternatively, an eigenvalue decomposition of the matrix $\Lambda(1 + i\gamma) + i\Phi^T K_s \Phi$ can be computed once for each excitation

frequency. But if $\Phi^T K_s \Phi$ is of very low rank, this fact can be exploited to reduce costs significantly, as will be shown in the next section.

The choice of the structural damping parameter γ is important to the rank of K_s. If the total structural damping matrix must be equal to $\gamma K + K_s$, but γ is allowed to vary from the predominant material's structural damping level, K_s can be modified to compensate for the change in γ. But this would increase the rank of K_s to that of K, eliminating an opportunity to reduce costs. Typically a human analyst chooses the value of γ, but some computer software receives the modal FRP that must be solved. If the matrix $\Phi^T K_s \Phi$ is received separately from matrix $\gamma\Lambda$, it is inexpensive to determine whether $\Phi^T K_s \Phi$ is of low rank. But if the two are combined together, their combination is typically a full-rank, fully populated symmetric matrix which is more expensive to handle in the modal FRP.

For this application, it is desirable to be able to find a "diagonal plus low rank" representation of matrix $\gamma\Lambda + \Phi^T K_s \Phi$ so that the most economical approach for solving the modal frequency response problem can be taken, regardless of whether the human analyst made the most advantageous choice of γ. Even if the rank of $\Phi^T K_s \Phi$ is not very low, in many cases damping levels are much lower than critical damping, so using an approximation consisting of a diagonal modal damping matrix and a full modal damping matrix of lower rank than $\Phi^T K_s \Phi$ might give acceptable accuracy at a lower cost.

1.1 Diagonal plus low rank approximation of symmetric matrices

The objective of this thesis is to find an approximation for a symmetric $n \times n$ matrix A such that

$$A \approx D + M \tag{1.5}$$

where D is diagonal and M is low rank of the form $V\Lambda V^T$, in which $V \in \mathbb{R}^{n\times p}$, $\Lambda \in \mathbb{R}^{p\times p}$, and p is minimized. The approximation is performed by minimizing the error measure

$$\alpha = ||A - \left(D + V\Lambda V^T\right)||_F{}^2, \tag{1.6}$$

where the subscript F stands for the Frobenius norm. The square of the Frobenius norm was chosen as a metric because it allows for minimizing the residual α term by term in V, Λ and D. The element-by-element representation of α is

$$\alpha = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(a_{ij} - d_i\delta_{ij} - \sum_{r=1}^{p} v_{ir}\lambda_r v_{jr}\right)^2 \tag{1.7}$$

The solution to find the matrix approximation in Eq. (1.5) involves finding matrices V, Λ and D and the number of columns p for matrices V and Λ. Matrix Λ is the diagonal matrix of non-zero eigenvalues of M, and V contains the corresponding eigenvectors. We refer to this method of representing symmetric matrices as a "diagonal plus low rank" (DPLR) approximation. For a matrix equal to $D + V\Lambda V^T$, with D diagonal and V and Λ of rank p, the

inverse can be obtained inexpensively using the Sherman-Morrison-Woodbury (SMW) formula:

$$\left(D + V\Lambda V^T\right)^{-1} = D^{-1} - D^{-1}V\left(\Lambda^{-1} + V^T D^{-1} V\right)^{-1} V^T D^{-1}. \tag{1.8}$$

This only requires diagonal D and a $p \times p$ matrix to be inverted, rather than an $n \times n$ matrix.

1.2 Thesis Outline

The following chapters take a look in depth at the process that led to obtaining optimal representations for matrices using the diagonal plus low rank (DPLR) approximation. Chapter 2 covers the theory behind DPLR. This chapter explains the implementation of different optimization methods to DPLR and the reason for choosing the Conjugate Gradient technique for obtaining the DPLR representation.

Chapter 3 discusses the procedure involved when dealing with the conjugate gradient algorithm. A detailed description of the DPLR algorithm is found here. This chapter also includes an operation count for the DPLR iterations.

Chapter 4 looks over the results obtained from the convergence of DPLR using different complex modal stiffness matrices. In this chapter we count the number of operations in various parts of the DPLR algorithm. We also develop an algorithm for solving linear systems of equations using iterative techniques. We compare two iterative techniques: one using the SMW

formula, and the other one using successive over-relaxation (SOR).

Chapter 5 summarizes conclusions drawn from the results and presents suggestions for future work related to DPLR approximations of matrices.

Chapter 2

Theory

The "diagonal plus low rank" (DPLR) approximation represents a symmetric matrix A as $A \approx D + M$ where D is diagonal and M is a low rank symmetric matrix of the form $V\Lambda V^T$. Here Λ is a diagonal matrix of non-zero eigenvalues of M, and V contains the respective eigenvectors. The error measure α that is minimized in order to obtain the optimal representation of A is given by

$$\alpha = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(a_{ij} - d_i\delta_{ij} - \sum_{r=1}^{p} v_{ir}\lambda_r v_{jr}\right)^2. \tag{2.1}$$

Three optimization approaches for minimizing α to obtain DPLR representations of matrices were considered, including the Newton-Raphson method, the steepest descent method and the conjugate gradient technique. These three optimization methods use the gradient of the error measure α. In addition to this, Newton-Raphson requires the calculation of the Hessian to iterate toward the optimal solution.

The gradient of α is given by

$$\boldsymbol{\nabla}\alpha = \sum \frac{\partial\alpha}{\partial x_k}\mathbf{e_k} \tag{2.2}$$

where x_k represents the variables: $d_i,\ i = 1, ..., n;\ \lambda_j,\ j = 1, ..., p$; and v_{kl}, $k = 1, ..., n,\ l = 1, ..., p$; which are the nonzero entries of matrices D, Λ and V respectively, and $\mathbf{e_k}$ is a cardinal direction of the space in which the gradient of α is being generated.

Minimizing α requires that $\boldsymbol{\nabla}\alpha = \mathbf{0}$, which requires partial derivatives of α with respect to variables d_i, λ_j and v_{kl} to be equal to zero. This gives several useful necessary conditions that must be satisfied.

Setting $\dfrac{\partial\alpha}{\partial d_k} = 0$ leads to

$$\frac{\partial\alpha}{\partial d_k} = -2\left(a_{kk} - d_k - \sum_{r=1}^{p} v_{kr}\lambda_r v_{kr}\right) = 0. \tag{2.3}$$

From Eq. (2.3) it can be observed that given V and Λ, α is minimized by choosing $d_k = a_{kk} - \sum_{r=1}^{p} v_{kr}\lambda_r v_{kr}$. This satisfies $\dfrac{\partial\alpha}{\partial d_k} = 0$ for any values in matrices V and Λ. With this choice of d_k, the diagonal terms in the matrix $A - \left(D + V\Lambda V^T\right)$ become zero. Setting $\dfrac{\partial\alpha}{\partial v_{kl}} = 0$ leads to

$$\frac{\partial\alpha}{\partial v_{kl}} = -4\lambda_l \sum_{j=1}^{n} \left(a_{kj} - \sum_{r=1}^{p} v_{kr}\lambda_r v_{jr}\right)_{e.d.} v_{jl} = 0, \tag{2.4}$$

and setting $\dfrac{\partial\alpha}{\partial\lambda_k} = 0$ gives

$$\frac{\partial\alpha}{\partial\lambda_k} = -2\sum_{k=1}^{n} v_{kl}\left(\sum_{j=1}^{n}\left(a_{kj} - \sum_{r=1}^{p} v_{kr}\lambda_r v_{jr}\right)_{e.d.} v_{jl}\right) = 0. \tag{2.5}$$

The subscript *e.d.* stands for "except the diagonal terms", which are eliminated by the choice of d_k's. Each of the λ_k's must be nonzero, because, if it is zero, it

and its corresponding column in V are simply discarded. We can see that Eq. (2.4) is embedded in Eq. (2.5), so we see that making $\frac{\partial \alpha}{\partial v_{kl}}$ equal to zero in Eq. (2.4) also makes $\frac{\partial \alpha}{\partial \lambda_k}$ equal to zero. Since the d_k's do not appear in Eq. (2.4), but are easily obtained from Eq. (2.3) once the v_{kl}'s and λ_k's have been found, we can focus our efforts on minimizing with respect to the v_{kl} and λ_k variables. In fact, if we temporarily allow vectors in V to be of arbitrary length rather than unit vectors, this has the same effect as allowing λ_k's to vary. So we can minimize α with respect to the entries in V, and then orthonormalize columns of V and update Λ by solving an eigenvalue problem, and finally solve for D by using Eq. (2.3). Reducing the number of minimization variables effectively to the entries in V simplifies the problem.

Setting $\boldsymbol{\nabla}\alpha = \mathbf{0}$ is necessary but may not be sufficient to achieve an accurate DPLR approximation that satisfies $A - \left(D + V\Lambda V^T\right) \approx 0$. If the rank p of matrices V and Λ is too small, the minimum value of α with this rank p will be non-zero. If α has been minimized with a given p, and is unacceptably large, we need to increase the number of columns p of matrix V. If $A - \left(D + V\Lambda V^T\right) = 0$ we need to make sure that the rank p of matrix V is optimal. To make sure that the rank p is minimized, we need to calculate α for a rank of $p-1$ and make sure that the new value of $\alpha|_{p-1}$ is smaller than the value of $\alpha|_p$.

The Newton-Raphson, steepest descent and conjugate gradient methods were investigated for obtaining DPLR approximations in order to determine which optimization technique is most efficient. An explanation of the

implementation of these minimization methods to obtain DPLR approximations is given below.

2.1 The Newton-Raphson method for DPLR approximations

It was observed that optimizing with respect to the entries of matrix V is sufficient to minimize α. By using the $()_{e.d.}$ operator we do not need to consider matrix D, and by letting the columns in matrix V be non-unit vectors we can let Λ stay constant without restricting the optimization. These facts allows us to only consider dependence on V. The Taylor series expansion of α with respect to V is

$$\alpha|_{\mathbf{v}+\mathbf{\Delta v}} = \alpha|_{\mathbf{v}} + \mathbf{\Delta v}^T \mathbf{\nabla} \alpha|_{\mathbf{v}} + \frac{1}{2} \mathbf{\Delta v}^T H|_{\mathbf{v}} \mathbf{\Delta v} + ..., \tag{2.6}$$

where vector $\mathbf{v} \in \mathbb{R}^{n \times p}$ is defined as

$$\mathbf{v} \equiv vec\,(V) = \left[\begin{array}{c} \mathbf{v_1}^T ... \mathbf{v_p}^T \end{array} \right]^T \tag{2.7}$$

in which the vector $\mathbf{v_i}$ is a column vector of matrix V. Matrix V is of size $n \times p$, where p is smaller than n. Matrix $H|_{\mathbf{v}}$ is the Hessian matrix related to the mixed second partial derivatives of metric α with respect to each entry in matrix V, evaluated at $\mathbf{v}$. The vector $\mathbf{\Delta v}$ is a change in $\mathbf{v}$.

When the derivative of $\dfrac{\partial \alpha}{\partial v_{kl}}$ is taken with respect to v_{xy} we obtain an

entry in H:

$$\frac{\partial}{\partial v_{xy}}\left(\frac{\partial \alpha}{\partial v_{kl}}\right) = -4\lambda_l \left(a_{kx} - \sum_{r=1}^{p} v_{kr}\lambda_r v_{xr}\right)_{e.d.} + 4\lambda_l\lambda_y \sum_{j=1}^{n} v_{jy}v_{jl}\delta_{kx} + 4\lambda_l\lambda_y v_{ky}v_{xl}, \tag{2.8}$$

The size of this matrix is $np \times np$. Separating each of the terms in Eq. (2.8) and organizing them in matrix form leads to the $np \times np$ matrices

$$E = 4\begin{bmatrix} \mathbf{v_1}^T\mathbf{v_1}I\lambda_1^2 & ... & \mathbf{v_p}^T\mathbf{v_1}I\lambda_p\lambda_1 \\ . & . & . \\ . & . & . \\ . & . & . \\ \mathbf{v_1}^T\mathbf{v_p}I\lambda_1\lambda_p & ... & \mathbf{v_p}^T\mathbf{v_p}I\lambda_p^2 \end{bmatrix},$$

where $\mathbf{v_j}$ represents a column vector of matrix V and matrix I is the identity matrix of size $n \times n$;

$$F = 4\begin{bmatrix} \mathbf{v_1}\mathbf{v_1}^T\lambda_1^2 & ... & \mathbf{v_p}\mathbf{v_1}^T\lambda_1\lambda_p \\ . & . & . \\ . & . & . \\ . & . & . \\ \mathbf{v_1}\mathbf{v_p}^T\lambda_1\lambda_p & ... & \mathbf{v_p}\mathbf{v_p}^T\lambda_p^2 \end{bmatrix},$$

in which the generic outer product $\mathbf{v_i}\mathbf{v_j}^T$ forms an $n \times n$ sub-matrix;

$$G = 4\begin{bmatrix} \mathrm{diag}(\mathbf{v_1}\mathbf{v_1}^T\lambda_1^2) & ... & \mathrm{diag}(\mathbf{v_p}\mathbf{v_1}^T\lambda_p\lambda_1) \\ . & . & . \\ . & . & . \\ . & . & . \\ \mathrm{diag}(\mathbf{v_1}\mathbf{v_p}^T)\lambda_1\lambda_p & ... & \mathrm{diag}(\mathbf{v_p}\mathbf{v_p}^T)\lambda_p^2 \end{bmatrix},$$

which has p^2 diagonal blocks of size $n \times n$; and

$$J = \text{block-diag}(J_1, ..., J_p) \tag{2.9}$$

where

$$J_i = 4\lambda_i^2 (A - V\Lambda V^T)_{e.d.}$$

is of size $n \times n$. The sum of $E + F - G - J$ is the Hessian H.

The net cost to form the Hessian matrix H is $O(np)^2$ flops per iteration, where p is less than n. The cost of calculating the gradient in Eq. (2.4) is $O(n^2 p)$ flops. In order to calculate the correction $\mathbf{\Delta v}$ using Newton-Raphson we must set the derivatives of α with respect to v_{kl}'s equal to zero, from Eq. (2.6). This results in the system of equations $H|_{\mathbf{v}} \mathbf{\Delta v} = -\mathbf{\nabla}\alpha|_{\mathbf{v}}$, which can be solved for the correction $\mathbf{\Delta v}$. To avoid the expense of factoring H, $\mathbf{\Delta v}$ can be found iteratively using the Jacobi or Gauss Siedel methods according to [5].

The fact that $(np)^2$ floating point operations per iteration are required to form matrix H makes the Newton-Raphson method increasingly expensive as the number of columns of matrix V increases. Since n is large, the order $(np)^2$ flops required while iterating using Newton-Raphson is larger than the order $n^2 p$ flops needed when iterating using the steepest descent or conjugate gradient methods described in the following two sections.

2.2 Steepest descent technique for DPLR approximation

The steepest descent technique minimizes α in the direction of the negative of the gradient in order to obtain a future iterate, as mentioned in

[5]. In the case of DPLR the correction is given by

$$\boldsymbol{\Delta}\mathbf{v} = -\gamma \boldsymbol{\nabla}\alpha, \tag{2.10}$$

where the search direction is the negative of the gradient, and is scaled by a step size γ. According to [5] and [6] the step size γ is given by

$$\gamma = \arg\min_{\gamma \geq 0} \alpha\left(\mathbf{v} + \gamma \boldsymbol{\Delta}\mathbf{v}\right) \tag{2.11}$$

The error measure α can be represented as a quartic polynomial in the step size γ. We can solve for the step size γ by taking the derivative of α with respect to γ and making $\dfrac{d\alpha}{d\gamma} = 0$. Then solving the cubic equation in γ yields the step size we are looking for. The representation of $\alpha\left(\gamma\right)$ is explained more thoroughly in the next section.

The steepest descent technique is the most cost-efficient when the function being minimized is quadratic. Then the maximum number of iterations to converge is equal to the number of unknowns. If the function α were a quadratic problem and well-conditioned, steepest descent would take at most np iterations to converge. The cost per iteration to calculate the correction $\boldsymbol{\Delta}\mathbf{v}$ using steepest descent is of order n^2p.

Problems with the convergence of the steepest descent method arise when the function being minimized is ill-conditioned and/or of higher order than quadratic, according to [6]. The error measure α is quartic in the entries in V. The method is not assured to converge in at most np iterations; hence, steepest descent might use search directions that are linear combina-

tions of previous search directions. This situation slows down the convergence of steepest descent to find a minimum.

Another issue observed concerning DPLR is the very flat narrow valley in which the minimizer is found. As we iterated using the steepest descent technique, we found that $||\mathbf{\Delta v}||_2$ was decreasing faster than the function α as we iterated using steepest descent. This leads the search direction to go in a "zig-zag" pattern towards the optimal solution when the search vector gets close to the desired minimum. This results in unnecessary iterations which make obtaining matrix V too expensive. Due to the previous two drawbacks, the steepest descent technique is not the most suitable approach for optimizing a DPLR approximation of a matrix.

Instead of following the gradient directions, in which two successive search directions are perpendicular to each other, a set of conjugate directions could be developed such that the new search directions are not linear combinations of the previous ones. According to [1] there exist conjugate direction techniques that allow the calculation of new search directions that are linearly independent during the iterative process. The conjugate gradient technique, described in the next section, is a very efficient approach for calculating search directions to minimize the error measure α.

2.3 Conjugate gradient method for DPLR approximations

The conjugate gradient technique is a very popular technique among gradient descent methods. One attractive attribute of this technique is that it can be easily adapted to solve nonlinear problems. The search direction for each iteration in this method is computed as

$$\beta = \frac{\boldsymbol{\nabla}\alpha|^T_{\mathbf{v}+\boldsymbol{\Delta}\mathbf{v}} \left(\boldsymbol{\nabla}\alpha|_{\mathbf{v}+\boldsymbol{\Delta}\mathbf{v}} - \boldsymbol{\nabla}\alpha|_{\mathbf{v}}\right)}{\boldsymbol{\nabla}\alpha|^T_{\mathbf{v}} \boldsymbol{\nabla}\alpha|_{\mathbf{v}}} \tag{2.12}$$
$$\boldsymbol{\Delta}\mathbf{v}_{k+1} = -\boldsymbol{\nabla}\alpha|_{\mathbf{v}+\boldsymbol{\Delta}\mathbf{v}} + \beta\boldsymbol{\Delta}\mathbf{v}|_k,$$

where β is the conjugacy factor that makes the new search direction conjugate (orthogonal with respect to the Hessian matrix H) to the previous search directions, $\boldsymbol{\Delta}\mathbf{v}_k$ is the current search direction, and $\boldsymbol{\Delta}\mathbf{v}_{k+1}$ is the new search vector. Initially we calculate the gradient of the error measure $\alpha|_{\mathbf{v}}$. Having calculated the gradient we set the first search direction to be $\boldsymbol{\Delta}\mathbf{v} = -\boldsymbol{\nabla}\alpha|_{\mathbf{v}}$. Then we calculate the step size γ that scales the search direction. We compute the conjugacy factor β described in Eq. (2.12). We calculate the search direction $\boldsymbol{\Delta}\mathbf{v}$ and finally we calculate the step size γ. The process enters an iterative loop in which successive corrections $\gamma\boldsymbol{\Delta}\mathbf{v}$ are calculated in order to minimize the error measure α.

According to [6] when the function α is not quadratic, we can hope for a certain level of conjugacy between the search directions $\boldsymbol{\Delta}\mathbf{v}$ with respect to the Hessian. However, conjugacy between search directions with respect to the Hessian is lost because the Hessian matrix H changes when calculated at

different $\mathbf{v}$'s.

To calculate the step size, we define ΔV to be an $n \times p$ matrix whose column vectors contain the entries of the search direction $\mathbf{\Delta v}$. Then α is

$$\alpha = || \left(A - (V + \gamma \Delta V) \Lambda (V + \gamma \Delta V) \right)_{e.d} ||_F^2. \tag{2.13}$$

The triple matrix multiplication $(V + \gamma \Delta V) \Lambda (V + \gamma \Delta V)$ inside the Frobenius norm can be expanded and this leads to

$$\alpha = || \left(A - V \Lambda V^T - \gamma \left(V \Lambda \Delta V^T + \Delta V \Lambda V^T \right) - \gamma^2 \Delta V \Lambda \Delta V^T \right)_{e.d} ||_F^2. \tag{2.14}$$

The square of the Frobenius norm of a real matrix $B \in \mathbb{R}^{n \times n}$ can be represented as

$$||B||_F^2 = \mathrm{tr} \left(B^T B \right), \tag{2.15}$$

where tr stands for the trace of a matrix. We can rewrite Eq. (2.15) and we obtain

$$||B||_F^2 = \sum_{i=1}^{n} \mathbf{b_i}^T \mathbf{b_i}, \tag{2.16}$$

where $\mathbf{b_i}$ is a column vector in B. The computation in Eq. (2.16) costs $O(n^2)$ flops. On the other hand, the computation in Eq. (2.15) costs $O(n^3)$ flops. If Eq. (2.14) is expanded using this approach, α becomes

$$\begin{aligned} \alpha \;=\; & \mathrm{tr} \left(A_{e.d.}^T A_{e.d.} \right) + \mathrm{tr} \left(S^T S \right) - 2\mathrm{tr} \left(A_{e.d.}^T S \right) + 2\gamma \mathrm{tr} \left(S^T S_d - A_{e.d.}^T S_d \right) \\ & + \gamma^2 \mathrm{tr} \left(S_d^T S_d - 2 A_{e.d.}^T S_{dd} + 2 S^T S_{dd} \right) \\ & + \gamma^3 2\mathrm{tr} \left(S_d^T S_{dd} \right) + \gamma^4 \mathrm{tr} \left(S_{dd}^T S_{dd} \right), \end{aligned} \tag{2.17}$$

where $S = (V\Lambda V^T)_{e.d.}$, $S_d = (V\Lambda\Delta V^T + \Delta V\Lambda V^T)_{e.d.}$, and $S_{dd} = (\Delta V\Lambda\Delta V^T)_{e.d.}$.

The function α in Eq. (2.17) is a quartic polynomial in γ. Therefore optimizing the step size is just a matter of obtaining a cubic polynomial from the derivative $\frac{\partial\alpha}{\partial\gamma}$, setting it equal to zero, and finding the root $\gamma > 0$ that minimizes α.

It is observed from the flop counts that the steepest descent and the chosen conjugate gradient techniques have about the same number of flops per iteration. The advantage of the conjugate gradient method for nonlinear functions, however, is that in the conjugate gradient iterations we obtain successive search directions that are conjugate with respect to the Hessian matrix H. This characteristic of conjugate gradient accelerates the convergence process by not going through directions that have already been explored, unlike steepest descent. The cost of this method is $O(n^2p)$ in every conjugate gradient iteration.

2.4 SMW based iterative solution to a linear system of equations

Linear systems of equations of the form $A\mathbf{x} = \mathbf{b}$ can be solved by using either direct methods or iterative techniques. When the DPLR representation is only an approximation, rather than being exact, we can use the DPLR approximation of symmetric matrices in order to solve a linear system of equations iteratively. Given the DPLR approximation of a matrix A as $A \approx D + V\Lambda V^T$, the inverse of this matrix approximation is given by the

Sherman-Morrison-Woodbury formula. The most expensive cost of the calculation in Eq. (2.18) is order n^2p, where $p < n$. Given the inverse of the matrix approximation, it is possible to solve a system of equations iteratively following a standard iterative solver approach given by

Algorithm 1: SMW-based algorithm to solve systems of linear equations.

Given coefficient matrix A, $D + V\Lambda V^T$
Define $\mathbf{x_0} = \mathbf{0}$, $k = 0$ and tolerance *tol* for SMW-based algorithm
Calculate residual $\mathbf{r_k} = \mathbf{b} - A\mathbf{x_0} = \mathbf{b}$
Define $A^\dagger = \left[D + V\Lambda V^T\right]^{-1}$ using the SMW formula
while $||\mathbf{r_k}|| > tol$ **do**
  Calculate $\Delta\mathbf{x_k}$: $\Delta\mathbf{x_k} = A^\dagger\mathbf{r_k}$
  Calculate new iterate $\mathbf{x_{k+1}}$: $\mathbf{x_{k+1}} = \mathbf{x_k} + \Delta\mathbf{x_k}$
  Calculate residual $\mathbf{r_{k+1}}$: $\mathbf{r_{k+1}} = \mathbf{b} - A\mathbf{x_{k+1}}$
  $k = k + 1$
end

Let us show the necessary and sufficient conditions required for this method to converge. To do so, we will show the first two iterations of the SMW-based algorithm. Given $\mathbf{x_0} = \mathbf{0}$, out initial iteration equation is

$$A\left(\mathbf{x_0} + \Delta\mathbf{x_0}\right) = \mathbf{b}. \tag{2.18}$$

Rearrenging, $A\Delta\mathbf{x_0} = \mathbf{b} - A\mathbf{x_0} = \mathbf{r_0}$. The correction $\Delta\mathbf{x_0}$ is computed

$$\Delta\mathbf{x_0} = A^\dagger\mathbf{r_0} = A^\dagger\mathbf{b}. \tag{2.19}$$

Calculate the new update $\mathbf{x_1} = \mathbf{x_0} + \Delta x_0$ which is equal to $A^\dagger\mathbf{b}$. For the next iteration we would like to satisfy $A\left(\mathbf{x_1} + \Delta\mathbf{x_1}\right) = \mathbf{b}$ so the correction $\Delta\mathbf{x_1}$ should satisfy $A\Delta\mathbf{x_1} = \mathbf{b} - A\mathbf{x_1} = \mathbf{r_1}$. This residual can be rewritten as

$$\mathbf{r_1} = \mathbf{b} - AA^\dagger\mathbf{b} = \left(I - AA^\dagger\right)\mathbf{b}. \tag{2.20}$$

The new correction $\mathbf{\Delta x_1}$ is calculated as $\mathbf{\Delta x_1} = \left(A^\dagger \mathbf{r_1} - A^\dagger A A^\dagger\right) \mathbf{r_1}$, which is also given by

$$\mathbf{\Delta x_1} = \left(I - A^\dagger A\right) A^\dagger \mathbf{b}. \tag{2.21}$$

After a couple of iterations in the SMW-based algorithm we obtain

$$\begin{aligned} \mathbf{r_k} &= \left(I - AA^\dagger\right)^k \mathbf{b} \\ \mathbf{\Delta x_k} &= \left(I - A^\dagger A\right)^k A^\dagger \mathbf{b}. \end{aligned} \tag{2.22}$$

This means that the convergence rate of the iterative technique is determined by how much smaller the spectral radius of the operator $\left(I - AA^\dagger\right)$ is than 1, which tells how much the residual $\mathbf{r_k}$ is reduced in every iteration. The iteration is convergent if the spectral radius is less than one.

Chapter 3

Methodology to Obtain DPLR Approximations for Symmetric Matrices

This chapter presents an algorithm for obtaining a DPLR approximation $A \approx D + M$ for symmetric matrices, where D is a diagonal matrix and M is a low rank symmetric matrix represented in terms of it eigenvalue decomposition $M = V\Lambda V^T$. The optimization problem is divided into two parts, explained in two different sections in this chapter. The first section explains the inner loop algorithm required to obtain matrices V and Λ given their number of columns p. The second section explains the outer loop calculations for determining whether the number of columns p is satisfactory or whether it needs to be increased or decreased.

3.1 Inner Loop

The purpose of the inner loop is to iteratively minimize the error measure α with a given number of columns p less than n by computing optimal V, Λ and D matrices. The iterations inside this loop follow the conjugate gradient algorithm, which is described in [5].

Algorithm 2: Inner Loop: Conjugate Gradient in DPLR.

Define $V_0 \in \mathbb{R}^{n \times p}$ coming from outer loop
Define $\mathbf{v_0}$ to be a np vector formed by the columns of matrix V_0
Let $tol_{||\mathbf{r_i}||}$, tol_α, and i_{max} be the tolerances for $\boldsymbol{\nabla}\alpha$, α and the maximum number of iterations in the conjugate gradient algorithm, respectively
Define κ_{tol} as the largest value that $\frac{d^2 \log \alpha}{di^2}$ can attain Calculate$\alpha_{\mathbf{v_0}}$ and $\boldsymbol{\nabla\alpha}_{\mathbf{v_0}}$
Set $\boldsymbol{\Delta}\mathbf{v_0} = -\boldsymbol{\nabla}\alpha_{\mathbf{v_0}}$, $\mathbf{r_0} = \boldsymbol{\Delta}\mathbf{v_0}$
while $||\mathbf{r_i}|| > tol_{||\mathbf{r_i}||}$ *and* $i < i_{max}$ *and* $\alpha > tol_\alpha$ **do**
  if $i \geq 3$ **then**
    Calculate second derivative of $\log \alpha_{i-1}$ **using central difference formula from finite difference methods**
    if $\kappa > \kappa_{tol}$ **then**
      Close while loop because $\frac{d^2 \log \alpha}{d^2 i} > 0$
    end
  end
  Arrange $\boldsymbol{\Delta}\mathbf{v_i}$ **into matrix form** ΔV_i**:**
    $\Delta V_i = \mathrm{mat}(\boldsymbol{\Delta}\mathbf{v_i})$
  Generate quartic α **polynomial from Eq. (2.13) as a function of the step size** γ_i
  Calculate step size γ_i **by solving** $\frac{\partial \alpha}{\partial \gamma_i} = 0$ **for** γ_i
  Calculate α **given** γ_i
  Calculate the value of V_{i+1} **for the next iteration:**
    $\mathbf{v_{i+1}} = \mathbf{v_i} + \gamma_i \boldsymbol{\Delta}\mathbf{v_i}$, $V_{i+1} = \mathrm{mat}(\mathbf{v_{i+1}})$
    $\mathbf{r_{i+1}} = -\boldsymbol{\nabla}\alpha_{\mathbf{v_{i+1}}}$
  Calculate conjugacy term β_i
  Calculate next search direction $\boldsymbol{\Delta}\mathbf{v_{i+1}}$**:**
    $\boldsymbol{\Delta}\mathbf{v_{i+1}} = \mathrm{mat}(-\boldsymbol{\nabla}\alpha_{\mathbf{v_{i+1}}} + \beta_i \boldsymbol{\Delta}\mathbf{v_i})$
  Check change in new matrix V**:**
    $v_d = ||\Delta V_{i+1}||_F^2$
  if $v_d < tol_{||\mathbf{r_i}||}$ **then**
    Stop convergence due to lack of improvement in matrix V
  else
    $i = i + 1$
  end
end

From Chapter two we know that

$$\alpha = ||(A - V\Lambda V^T - \gamma(V\Lambda\Delta V^T + \Delta V\Lambda V^T) - \gamma^2\Delta V\Lambda\Delta V^T)||_F^2. \quad (3.1)$$

Define $C_0 = A - V\Lambda V^T$, $C_1 = V\Lambda\Delta V^T + \Delta V\Lambda V^T$ and $C_2 = \Delta V\Lambda\Delta V^T$. Then expanding the Frobenius norm in Eq. (3.1) we obtain

$$\begin{aligned} \alpha &= \mathrm{tr}(C_0^T C_0 - \gamma(C_0^T C_1 + C_1^T C_0) - \quad (3.2) \\ &\gamma^2(C_0^T C_2 + C_2^T C_0 - C_1^T C_1) \\ &\quad +\gamma^3(C_1^T C_2 + C_2^T C_1) + \gamma^4(C_2^T C_2)), \end{aligned}$$

Since $\mathrm{tr}\,(X + Y) = \mathrm{tr}\,(X) + \mathrm{tr}\,(Y)$, Eq. (3.2) becomes

$$\begin{aligned} \alpha &= \mathrm{tr}(C_0^T C_0) - 2\mathrm{tr}(C_1^T C_0)\gamma + \mathrm{tr}(C_1^T C_1 - 2C_0^T C_2)\gamma^2 + \quad (3.3) \\ &\quad 2\mathrm{tr}(C_1^T C_2)\gamma^3 + \mathrm{tr}(C_2^T C_2)\gamma^4. \end{aligned}$$

The expression in Eq. (3.3) is a scalar polynomial that is quartic in γ, which can also be written as

$$\alpha(\gamma) = c_0 - 2c_1\gamma + c_2\gamma^2 + 2c_3\gamma^3 + c_4\gamma^4. \quad (3.4)$$

Setting $\dfrac{\partial\alpha}{\partial\gamma} = 0$ leaves us with a cubic equation to be solved for the unknown γ, so the calculation of the step size associated with the search direction is simple and easy to calculate.

In order to obtain a conjugate direction, the conjugate gradient update parameter β_i has to be calculated. When the function being minimized is quadratic ($f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} + \mathbf{b}^T\mathbf{x} + c$), this conjugate parameter makes every

search direction conjugate to the previous ones with respect to the coefficient matrix A. On the other hand, if the function is of higher degree, the β_i term will make each search direction approximately conjugate to previous ones with respect to the matrix of mixed second derivatives of the function being minimized, according to [6] and [4]. In our case the function being minimized is the error function $\alpha(\text{vec}(V))$. Different formulas have been developed for calculating the conjugate parameter β_i. According to [2], using alternate formulas in each iteration to calculate the β_i term often improves convergence of the conjugate gradient algorithm. The formulas used to find β_i in the development of the DPLR algorithm are

$$\begin{aligned} \beta_i^{PRP} &= \frac{\boldsymbol{\nabla}\alpha_{i+1}^T(\boldsymbol{\nabla}\alpha_{i+1} - \boldsymbol{\nabla}\alpha_i)}{||\boldsymbol{\nabla}\alpha_i||^2} \\ \beta_i^{FR} &= \frac{||\boldsymbol{\nabla}\alpha_{i+1}||^2}{||\boldsymbol{\nabla}\alpha_i||^2}. \end{aligned} \tag{3.5}$$

We choose β_i such that $\beta_i = \max\left(0, \min\left(\beta^{PRP}, \beta^{FR}\right)\right)$, where superscripts PRP and FR stand for Polak, Ribiere and Polyak [2], and Fletcher and Reeves [2], respectively.

3.2 Outer Loop

The outer loop's task is to determine the minimum rank p of matrices V and Λ such that the error measure α is smaller than tol_α. If a system of equations is being solved iteratively, tol_α must be small enough that the iterative approach converges acceptably quickly. The outer loop's algorithm is shown next:

Algorithm 3: Outer loop algorithm: Defining rank p for V and Λ.

Given symmetric $A \in \mathbb{R}^{n \times n}$:
Define $V, \Lambda \in \mathbb{R}^{n \times n}$ such that $V = \Lambda = I$
Define initial $p = \lfloor \frac{n}{10} \rfloor$
Define α_{rtol} to be the tolerance for $e_0 = \frac{\alpha|_{\mathbf{v}}}{||A||_F^2}$
Set $k = 0$
Select initial guess coming from the first p columns of matrix V and call it V_{p_k}
$i = p$
while $i < n$ *and* $e_k > \alpha_{rtol}$ **do**
  Optimize V_{p_k} **and** α_k **using the inner loop algorithm**
  Calculate e_k
  if $e_k > \alpha_{rtol}$ **then**
    Orthonormalize V_{p_k} **and update** Λ_{p_k} **Determine new rank** p_{k+1} **of matrices** $V_{p_{k+1}}$ **and** Λp_{k+1} **by truncating** Λ_{p_k}
    if $p_{k+1} < p$ **then**
      Optimize matrix $V_{p_{k+1}}$ **and** α_{k+1} **using the inner loop algorithm**
    end
  else
    $p_{k+1} = p_k + \lfloor \frac{n}{10} \rfloor$
  end
  $k = k + 1$
  $i = i + p_k$
end
Orthonormalize final matrix V_{p_k} **and update** Λ_{p_k}

Algorithm 3 shows how the choice of the rank p for matrices V and Λ is performed. As mentioned earlier, matrix V is not required to remain orthonormal (so that $V^T V = I$) during the conjugate gradient iterations, but it can be orthonormalized through the solution of a generalized eigenvalue problem. To orthonormalize matrix V, let us first define V_{CG} to be V calculated using the

conjugate gradient technique. Let us also define $C \equiv V_{CG}^T V_{CG}$ and $B \equiv \Lambda^{-}1$. Then the eigenvalue problem is defined as

$$B\Phi = C\Phi\Upsilon, \tag{3.6}$$

so that $\Phi^T B\Phi = \Upsilon$ and $\Phi^T C\Phi = I$. We can see that $B^{-1} = \Phi\Upsilon^{-1}\Phi^T$. Let $V_N \equiv V_{CG}\Phi$ and $\Lambda_N \equiv \Upsilon^{-1}$, then we can verify that $V_N^T V_N = I$ and $M = V_N \Lambda_N V_N^T = V_{CG}\Lambda V_{CG}^T$. Matrix V_N is the orthonormalized matrix V and Λ_N is the updated matrix Λ. The solution of this eigenvalue problem costs $O(p^3)$ flops.

3.3 Floating Point Operation Count in the DPLR Algorithm

The floating point operation (flop) count in the DPLR algorithm varies due to the continuous change of rank p of matrix V. The most computationally expensive steps in the DPLR algorithm take place in the conjugate gradient iteration. A detailed flop count is given in Table 3.1.

Operation	**Flop count**
Gradient	$3n^2p + 4np^2$
Form polynomial	$6n^2p + 10n^2 - 6n$
Calculating new matrix V	$2np$
Calculation of conjugate gradient formula	$7np - 3$
Calculation of new search direction	$2np$
Checking variation ΔV	$2np$
Total number of flops	$9n^2p + 10n^2 + 4np^2$

Table 3.1: Flop count per iteration inside the conjugate gradient algorithm.

The costs found in Table 3.1 are costs per iteration inside the conjugate gradient algorithm. To these costs we have to add the costs involved in the outer loop. They add up to $O(n^2p + p^3)$.

It is important to point out that the flop counts in Table 3.1 are the costs of only one iteration to obtain a DPLR matrix approximation. The total flop count depends on the number of iterations that conjugate gradient takes to converge to the optimal matrix V.

Chapter 4

Results

In this chapter the results of obtaining a DPLR approximation of a symmetric matrix are presented. The accuracy of the method is investigated, and convergence of the conjugate gradient technique used to obtain a DPLR approximation is examined. Studies on solving linear systems of equations using a new iterative technique based on the SMW formula are performed.

First we take a look at the accuracy of the DPLR algorithm when approximating a matrix which is equal to a diagonal matrix plus a low rank symmetric matrix. This test is performed in order to validate the effectiveness of the DPLR algorithm. After performing the validation of the DPLR algorithm, we approximate a symmetric matrix taken from a test case from industry using the DPLR approach. This test is performed in order to study the accuracy of DPLR when it approximates matrices that are not known to be the sum of a diagonal matrix and a low rank matrix.

After studying the ability of the DPLR approach to approximate symmetric matrices, we focus on the solution of linear systems of equations using the representation of matrices obtained from the DPLR algorithm. We investigate performance in solving the modal FRP of a structure, whose governing

equation is

$$\left(-\omega^2 I + \Omega(1 + i\gamma) + i\bar{K}_s\right) \boldsymbol{\eta} = \mathbf{b}, \tag{4.1}$$

where ω^2 is the square of an excitation frequency, Ω is a diagonal matrix containing the squares of the natural frequencies of the structure below the cutoff value, γ is the structural damping parameter for the predominant material in the structure, and $\bar{K}_s$ is the modal structural damping matrix representing the difference from γ for the other materials in the structure. Using the DPLR algorithm we can represent the entire modal structural damping matrix in the form $\gamma\Omega + \bar{K}_s \approx D + V\Lambda V^T$. Then using the SMW-based algorithm we can solve the modal FRP for multiple excitation frequencies ω and multiple right hand sides.

In order to test the DPLR algorithm we used two matrices: A_B, and A_I. Matrix A_B is a symmetric matrix built as $A_B = D_B + M_B$, where matrix D_B is diagonal and matrix M_B is a low rank symmetric matrix constructed from the product $\Phi_B \Lambda_B \Phi_B^T$ where $\Phi_B^T \Phi_B = I$, Φ_B is of size 600×58. The values of Λ_B are chosen to be spread apart, and D_B, Phi_B and Λ_B are populated with random entries.

Matrix A_I is a sub-matrix which comes from a modal structural damping matrix that was generated in the car industry. The size of the matrix A_I is 600×600.

4.1 DPLR application to a diagonal plus low rank symmetric matrix

In order to evaluate the accuracy of the algorithm developed for obtaining a DPLR approximation of a matrix, different test cases for approximating matrices $A \equiv D + M$, where D is a diagonal matrix and M is a low rank symmetric matrix, were performed. We show a graphical representation of the contents of a test matrix A in Fig. 4.1. Two other graphs per test matrix were generated. Fig. 4.2 shows the error matrix $E = A - \left(D + V\Lambda V^T\right)$, and Fig. 4.3 shows the convergence of the DPLR algorithm.

The first case is that of matrix A_B. This matrix has its largest entries on its diagonal. Figure 4.1 is a graphical representation of A_B and it was generated using Matlab. The color spectrum allows us to identify the largest entries in magnitude in each of the generated graphs that represent a matrix. In the case of matrix A_B, we can observe a yellow-orange diagonal line going across the matrix representation. The rest of the matrix has light blue tones. Matrix A_B is built to have its largest values along its diagonal since this is a typical characteristic of matrices that would be encountered if the DPLR algorithm were implemented for use in the industry.

Matrix A_B is approximated as $D + V\Lambda V^T$ using the DPLR algorithm. To verify the accuracy of the DPLR algorithm we calculate the error matrix $E = A_B - \left(D + V\Lambda V^T\right)$. In Fig. 4.2 we observe the graph of the error matrix. From Fig. 4.2 it can be said that the DPLR approximation performed an accurate solution representing matrix A_B. Note that the color spectrum in Fig.

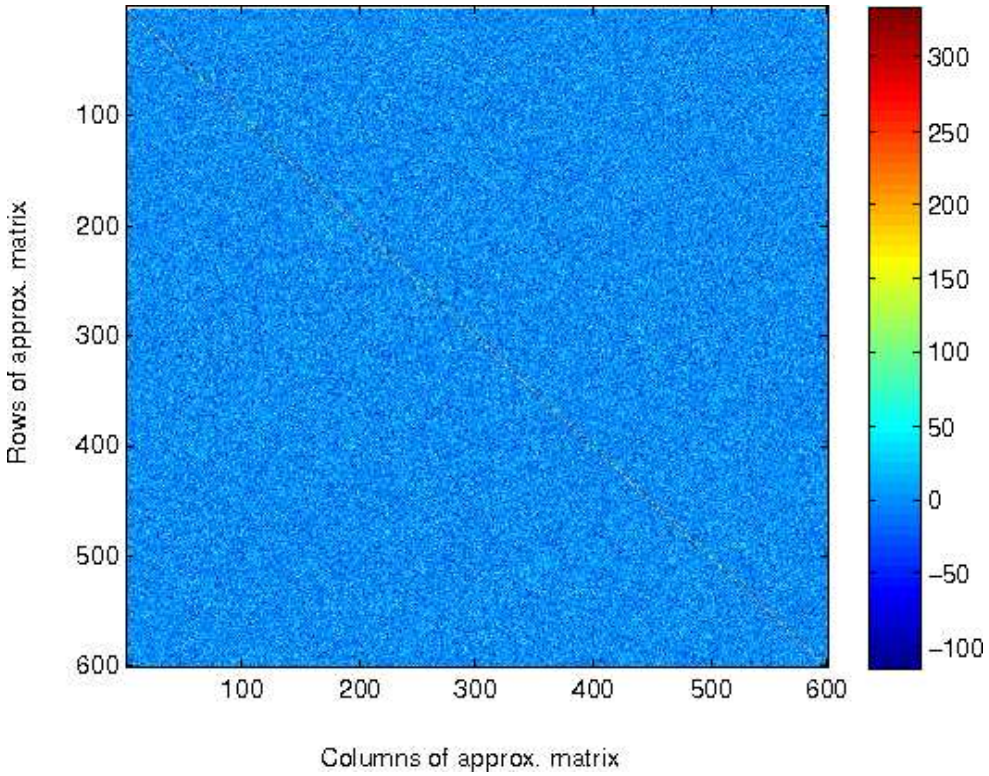

Figure 4.1: Graphical representation of a matrix A_B.

4.2 represents much smaller values than the ones represented in the color spectrum for Fig. 4.1. The error matrix, E, shows that the DPLR approximation algorithm has represented matrix A_B satisfactorily.

To examine the results in Fig. 4.2 more deeply, we plot the relative error $\frac{||\left(A_B - V\Lambda V^T\right)_{e.d.}||_F^2}{||A||_F^2}$ as the number of conjugate gradient iterations increases in Fig. 4.3.

The initial rank p for matrices V and Λ is 20. The iteration process is split into four parts. The first part includes the first 12 iterations. We can

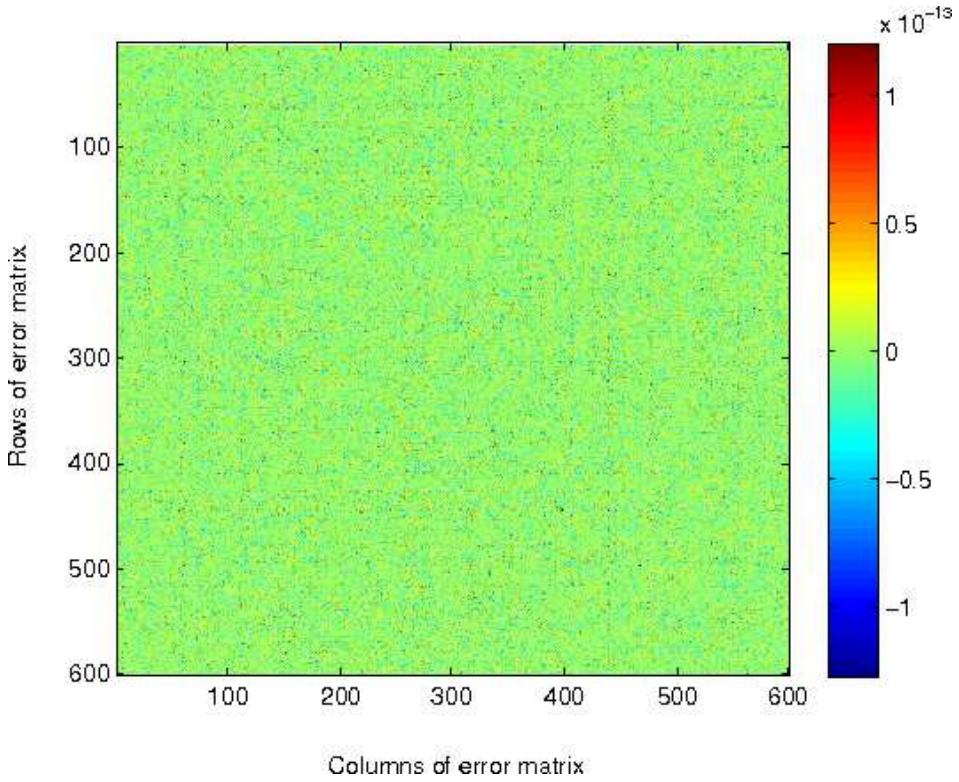

Figure 4.2: Error matrix graphical representation for matrix A_B.

see that the gradient is slowly decreasing. Given this drawback, the DPLR algorithm increases the rank of matrices V and Λ up to 40. We can observe a small improvement in the convergence of DPLR algorithm with a rank of 40. Then the rank of matrices V and Λ is increased up to 60 when the number of iterations is 30. There is a noticeable improvement in the convergence of the DPLR algorithm we reach iteration 46. At this point the slope of the relative error starts decreasing as observed in Fig. 4.3. At iteration 50 the DPLR algorithm truncates the rank of matrices V and Λ down to 58 and the exact

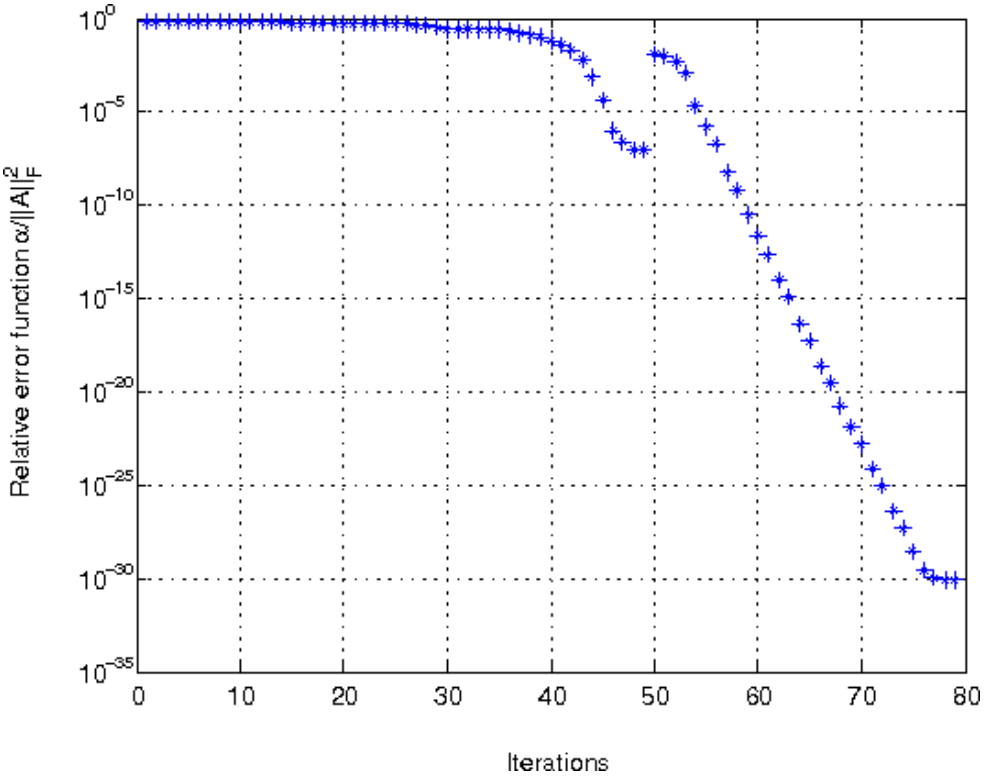

Figure 4.3: Convergence of conjugate gradient to find a DPLR approximation for matrix A_B

DPLR representation of matrix A_B is obtained. The lowest value that the relative error reaches is $\frac{\alpha}{||A_B||_F^2}$ of 9.5564×10^{-28}. Each entry inside the error matrix E is less than $\sqrt{\alpha}$ which is $O\left(10^{-12}\right)$ and very small as seen in Fig. 4.2. Hence we state that the DPLR approximation of matrix A_B is accurate.

Now that we have demonstrated that we can accurately represent a diagonal matrix plus rank 58 symmetric matrix, we can study general cases in which the matrices A to be represented are not known to come from the sum

of a diagonal matrix plus low rank symmetric matrix.

4.2 DPLR application to a general symmetric matrix

Since we have seen that the DPLR approximations algorithm can accurately represent a symmetric matrix of the form $A = D + M$, where matrix D is diagonal and matrix M is low rank symmetric, we now take a look at a more generalized scenario in which we only know that the matrix to be approximated is symmetric.

Let us study the approximation of a matrix $A_I \in \mathbb{R}^{600\times600}$ that is known to be diagonally dominant, as observed in Fig. 4.4.

The data in matrix A_I increases in magnitude as one traverses through the matrix from the top left corner to the bottom right corner, as observed in Fig. 4.4.

Matrix A_I is then approximated using the DPLR algorithm. The convergence criterion for this case drives the relative error $\frac{\alpha}{||A_I||_F^2}$ below the value of 1×10^{-2}. It was observed that when the relative error is smaller than 1×10^{-2} we obtain an acceptable representation of A_I, that allows us to solve the modal FRP. The accuracy of the method is highlighted when we take a look at the error matrix $E = A_I - \left(D + V\Lambda V^T\right)$. The graphical representation of matrix E can be observed in Fig. 4.5. The color spectrum in Fig. 4.5 shows us that almost all the entries in matrix A_I are $O(10^1)$. We can observe some matrix entries in E that are $O(10^3)$. Even though this error is large, the relative error

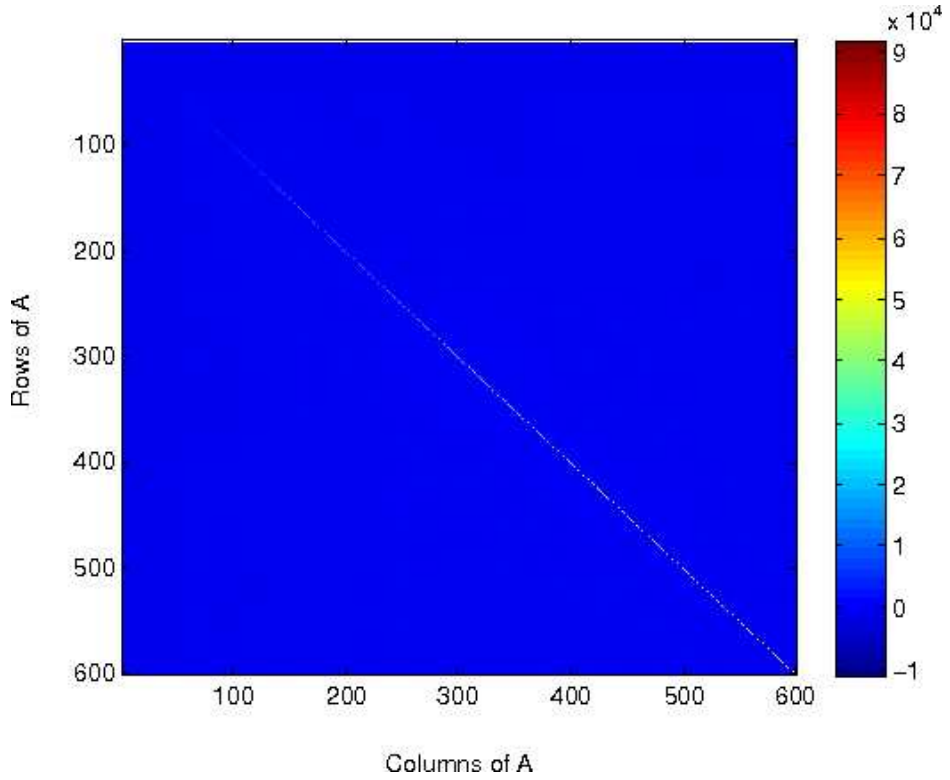

Figure 4.4: Graphical representation of a symmetric matrix A_I.

is small enough to provide us with an satisfactory DPLR approximation of matrix A_I so we can use this DPLR approximation to solve the modal FRP of a structure.

The convergence of the conjugate gradient iterations to obtain a DPLR approximation for matrix A_I can be observed in Fig. 4.6. The initial rank of matrices V and Λ is 20. It can be seen in Fig. 4.6 that the relative error $\frac{\alpha}{||A_I||_F^2}$ presents a monotonic decrease with respect to the number of iterations for a given rank p. A small jump in the relative error can be observed between

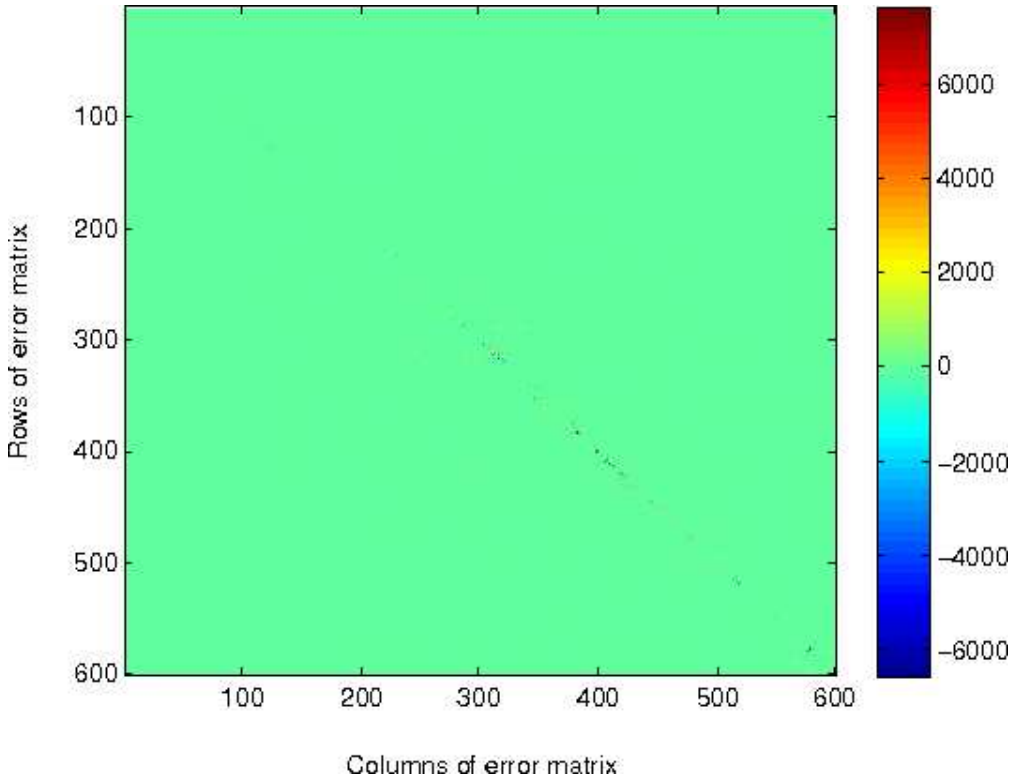

Figure 4.5: Graphical representation of the error matrix $E = A_{I_1} - \left(D + V\Lambda V^T\right)$.

iterations 3 and 4. This jump happens because the rank of matrices V and Λ is increased. Given the number of columns in V and Λ, which is the rank of matrix $M = V\Lambda V^T$, we can expect the conjugate gradient iterations to minimize α such that the error measure α approaches its global minimum given the rank of M. The DPLR approximation algorithm was stopped when the relative error $\frac{\alpha}{||A||_F^2}$ was less than 1×10^{-2}, leaving V and Λ with 39 columns. Increments in the ranks of matrices V and Λ were performed when the slope of $\log \alpha$ with respect to the number of iterations became too small.

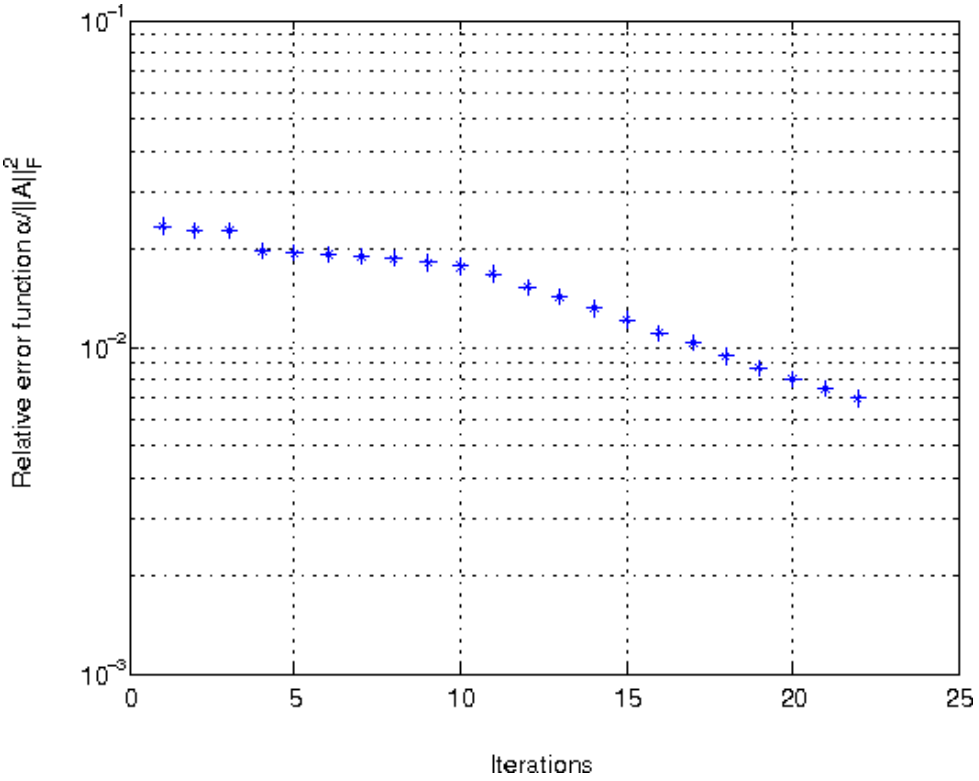

Figure 4.6: Conjugate gradient convergence in DPLR algorithm to approximate matrix A_I.

4.3 Solving modal frequency response problems using DPLR matrix approximations

The DPLR approximation of matrices is meant to approximate part of the symmetric coefficient matrix in the FRP

$$\left(-\omega^2 I + \Omega + iA\right)\boldsymbol{\eta} = \mathbf{b}, \tag{4.2}$$

where A is the fully populated matrix being approximated through the DPLR algorithm. Two basic approaches for solving a linear system of equations are

using direct methods or iterative techniques. The DPLR algorithm allows us the opportunity to solve a linear system of equations using both approaches. When matrix A in Eq. (4.2)is given exactly by $A = D + M$, where D is a diagonal matrix and matrix M is symmetric low rank, then we can solve this equation directly by using the SMW formula described in Chapter 1 of this thesis.

There will be an error associated with having only a DPLR approximation of matrix A using the direct approach. Since the convergence of the DPLR algorithm is specified by a tolerance, there is an error involved when solving a system of equations using the approximation of matrix A. However, as we will see later in this section, when solving the system of equations in Eq. (4.2), the residual satisfies the tolerance imposed on the SMW-based algorithm after one iteration. Hence we say that the solution of the system of linear equations is direct.

When we have a modal structural damping matrix that is not a diagonal matrix plus a low rank symmetric matrix, we must solve the system of linear equations iteratively since we can only obtain a DPLR approximation of the structural damping matrix A in Eq. (4.2). In this section we look at the total number of floating point operations required to solve the modal frequency response problem in Eq. (4.2).

The DPLR algorithm and the SMW-based algorithm are meant to be useful in solving the modal response problem of a structure, where the value ω^2 is the square of an excitation frequency. We will need the SMW-based iterative

technique to solve the modal response problem for multiple excitation frequencies ω and multiple right hand sides $\mathbf{b}$, with only one DPLR approximation of matrix A.

Concerning the iterative technique, we look at the stability of the SMW-based algorithm to solve linear systems of equations. We also examine the number of floating point operations required to solve iteratively the system of equations in Eq. (4.2) using different matrix sizes.

4.3.1 Solution to modal frequency response problem possessing a diagonal matrix plus low rank symmetric structural damping matrix

The first case tested is a symmetric matrix $A_B \in \mathbb{R}^{600\times 600}$, described at the beginning of this chapter. Tables 4.1 and 4.2 show the performance of DPLR and SMW to solve the system of linear equations in Eq. (4.2).

Number of flops in DPLR	1.2559E+10
Relative error in DPLR	8.1187E-31

Table 4.1: Performance to DPLR approximate the modal structural damping matrix A_B.

Number of iterations in SMW	1
Number of flops in SMW	6.9883E+07
Residual norm in SMW	2.5705E-16
Relative residual in SMW	2.1325E-19

Table 4.2: Performance of SMW-based algorithm solving a modal response problem of order 600 with modal structural damping matrix A_B.

Table 4.1 shows the performance of the DPLR algorithm in obtaining the DPLR approximation of matrix A_B. The relative error in DPLR is given by $\frac{\alpha}{||A_B||_F^2}$. The DPLR algorithm is performed only once to represent A_B. Table 4.2 shows the performance of the SMW-based algorithm to solve a system of linear equations which could have multiple right hand sides. The number of flops shown in Table 4.2 is the number of flops required to solve the modal FRP for each right hand side. The relative residual to solve system of linear equations is given by $\frac{||\mathbf{b} - (-\omega^2 I + \Omega + iA_B)\,\mathbf{x}^*||_2}{||\mathbf{b}||_\mathbf{2}}$, where $\mathbf{x}^*$ is the solution found to the system of equations in Eq. (4.2). The fact that the error measure α in the DPLR algorithm is not forced to be exactly equal to zero leads to having an error that will propagate to the solution of Eq. (4.2), which is observed in the residual norm in SMW in Table 4.2. The solution of the modal FRP is satisfactory using the SMW-based algorithm if we consider that the relative residual in SMW is $O(10^{-19})$. Therefore we can trust this method for solving systems of linear equations.

In the case presented previously, A_B had a low rank matrix contribution whose rank was less than ten percent of the dimension of A_B. We chose to work with such a low rank matrix because there are cases in which the complex structural damping matrix has such low rank characteristics, as explained in Chapter 1.

When solving the FRP we prescribed the range of the square of the excitation frequencies to be between $3000\frac{rad^2}{s^2}$ and $3300\frac{rad^2}{s^2}$. We chose this range for the square of the excitation frequencies because the square of the

lowest natural frequency contained in matrix Ω is $3204.6985 \frac{rad^2}{s^2}$. We wanted to observe whether a second or higher iteration would be necessary if the square of the excitation would be close to the square of lowest natural frequency. After running the SMW-based algorithm to solve the modal response problem for all the prescribed excitation frequencies ω it was observed that the flop cost per excitation frequency was 6.9883×10^8 flops, which is equivalent to one iteration in the SMW-based algorithm. The flop count was not affected by the range of excitation frequencies being close to the lowest natural frequency of vibration of the structure.

4.3.2 Solution to a modal frequency response problem possessing a general symmetric structural damping matrix

We now shift our attention to the solution of the modal FRP in Eq. (4.2) for matrices in which the complex structural damping matrix A is not known to be formed by adding a diagonal matrix plus a low rank symmetric matrix. In this case we can only approximate the modal structural damping matrix. Let us substitute matrix A in Eq. (4.2) with matrix A_I, described earlier in this chapter. Now we can observe at the performance of the DPLR algorithm when approximating the modal structural damping matrix A_I in Table 4.3. Table 4.4 shows the flop count when solving the modal frequency response problem in Eq. (4.2).

The DPLR algorithm returned an approximation of matrix A_I given by a diagonal matrix plus a rank 39 matrix. The convergence of the DPLR

Number of flops in DPLR	2.7647E+09
Relative error in DPLR	7.0767E-03

Table 4.3: Performance in DPLR approximating a modal structural damping matrix A_I.

Number of iterations in SMW	7
Number of flops in SMW	2.9747E+08
Residual in SMW	4.9000E-06
Relative residual in SMW	1.3809E-08
Relative tolerance in SMW	1.0000E-06

Table 4.4: Performance of SMW-based algorithm solving a modal frequency response problem with modal structural damping matrix A_I.

algorithm is determined by the relative error $\frac{\alpha}{||A_I||_F^2}$. If the relative error is less that 1×10^{-2}, then we are assured a satisfactory representation of matrix A_I to solve the modal response problem. If we wanted a lower relative error we would have to increase the rank of matrices V and Λ that DPLR approximate A_I. From Table 4.4 we can see that the solution of the modal response problem is iterative. This happens because the modal structural damping matrix is not represented exactly using the DPLR algorithm. The error from the DPLR algorithm then propagates to the solution of the modal response problem when using the SMW-based algorithm.

When solving the modal FRP we observe that the number of flops increases starting at $\omega^2 = 3128\frac{rad^2}{s^2}$. This is related to the fact that we have modified the spectral radius of the operator that indicates the rate of convergence of the SMW-based algorithm. This operator is given by $I - TT^{\dagger}$, where

$T \equiv -\omega^2 I + \Omega + iA_I$ and $T^\dagger \equiv -\omega^2 I + \Omega + i\left(D + V\Lambda V^T\right)$. The increase in the number of flops also indicated the increment in the number of iterations to solve the FRP.

4.4 Comparing iterative techniques to solve modal frequency response problems

This section compares the performance characteristics of the SMW-based iterative technique discussed in chapter two of this thesis to the performance of the Successive Over-Relaxation technique, which is a solver frequency used to solve linear systems of equations having positive definite coefficient matrices. We first take a look at the solution of the modal response problem in Eq. (4.2) having A_B modal structural damping matrix. We solve Eq. (4.2) using the SMW-based algorithm and SOR technique. Then we solve another modal response problem having A_I as the modal structural damping matrix.

4.4.1 Comparing iterative techniques to solve modal frequency response problems possessing structural damping matrices of the form diagonal matrix plus low rank symmetric matrix

Let us take a look at the flop count performance of the solution of modal frequency response problems in Eq. (4.2). Let us replace matrix A in Eq. (4.2) with A_B. This makes the solution of the modal response problem to be of dimension equal to 600. Tables 4.5 and 4.7 show the flop counts required to solve a system of equations. In the case of the SMW-based algorithm, we include the flops required to obtain a DPLR approximation of the coefficient

matrix A when solving Eq. (4.2).

Number of flops in DPLR	1.2559E+10
Relative error in DPLR	8.1187E-31

Table 4.5: Performance in DPLR approximating a modal structural damping matrix A_B .

Number of flops in SMW	6.9883E+07
Total number of iterations in SMW	1
Error in SMW	2.5705E-16
Relative error in SMW	1.9486E-19

Table 4.6: Performance solving a modal response problem using the SMW-based algorithm with modal structural damping matrix A_B

We can observe that the most expensive part of solving the modal response problem is finding the DPLR approximation of the modal structural damping matrix A_B. Recall that the DPLR approximation is performed just once, and the iterative solution is carried out once per right hand side, per excitation frequency.

We now observe the performance of the SOR algorithm in solving the same system of equations, which is summarized in Table 4.7.

The SOR algorithm does not have to compute an approximation of the coefficient matrix A_B which makes its total number of flops required to solve the system of equations smaller than the total number of flops required to solve the same system of equations using the SMW-based algorithm for few right hand sides and few excitation frequencies. If we focus only on the solution

Number of flops in SOR	3.16414E+09
Total number of iterations in SOR	548
Error in SOR	9.6945E-06
Relative error in SOR	8.0427E-09
Absolute Tolerance in SOR	1.0000E-06

Table 4.7: Performance solving a modal response problem with leading dimension of 600

of the system of equations, without taking into account the need to compute an approximation for A_B in the SMW-based algorithm, we observe that the SMW-based algorithm costs less flops than the SOR technique. It would be helpful to know if the DPLR approximation algorithm and the SMW-based technique can together cost less flops than the SOR technique. Since we want to solve the frequency response problem for multiple excitation frequencies and multiple right hand sides, we can solve the inequality

$$C + (XY)S \geq (XY)O, \tag{4.3}$$

and determine the combination of number of right hand sides and number of excitation frequency required to outperform the SOR technique. In Eq. (4.3) C stands for the number of flops in DPLR, S is the number of flops in the SMW algorithm, O is the number of flops SOR, XY is the number of combinations between the number of excitation frequencies, X, and the number of right hand sides, Y, required by the SMW-based algorithm and DPLR approximation algorithm to have a better flop performance than SOR has. Solving for XY we obtain that $XY \geq \dfrac{C}{0-S}$. For this particular case we

need the XY to be greater than four, i.e. we could solve the modal frequency response problem for three excitation frequencies and two right hand sides for the SMW-based algorithm and the DPLR approximation technique to cost less than the SOR technique.

4.4.2 Comparing iterative techniques to solve modal response problems possessing general symmetric modal structural damping matrices

Now we concentrate our studies in analyzing the performance of the DPLR algorithm and the SMW-based iterative technique in solving linear systems of equations for general symmetric matrices. We consider the modal FRP in Eq. (4.2) and we approximate the modal structural damping matrix A_I using the DPLR approximation algorithm. Let us take a look at the performance of the DPLR and SMW-based algorithms in Tables 4.8 and 4.9 when representing matrix A_I and solving the modal FRP respectively. The dimension of the modal response problem is 600.

Number of flops in DPLR	2.7647E+09
Number of columns of matrix V in DPLR	39
Relative error in DPLR	7.07674E-09

Table 4.8: Performance in DPLR approximating a general symmetric modal structural damping matrix A_I

The relative error in DPLR in Table 4.9 is given by $\frac{\alpha}{||A||_F^2}$. The most computationally intensive calculation when solving the system of equations using the SMW-based algorithm is the calculation of the approximation of the

Number of flops in SMW	2.9747E+08
Total number of flops DPLR+SMW	3.0622E+09
Total number of iterations in SMW	7
Error in SMW	4.9000E-06
Relative error in SMW	1.3809E-08
Relative tolerance in SMW	1.0000E-06

Table 4.9: Performance solving a modal response problem using the SMW-based algorithm for linear system with a general symmetric modal structural damping matrix A_I.

modal structural damping matrix A_I. In Table 4.10 we can see the performance of the SOR algorithm when solving the frequency response problem in Eq. (4.2) Comparing the number of flops between the SOR algorithm and

Number of flops in SOR	3.1410E+08
Total number of iterations in SOR	53
Error in SOR	9.2238E-06
Relative error in SOR	2.5993E-08
Absolute tolerance in SOR	1.0000E-06

Table 4.10: SOR performance solving modal frequency response problem having modal structural damping matrix A_I.

the SMW-based technique, we observe that the flop count in SOR is of the same order of magnitude as the one in the SMW-based algorithm. The relative error resulting from the SMW-based algorithm is more accurate than the one found using SOR. If one were to compare these specifications coming from the two algorithms, the SOR algorithm would be chosen over the SMW-based algorithm because the SOR algorithm is computationally cheaper than the SMW-based technique. The solution of the modal FRP costs more for

few right hand sides and few excitation frequencies when solving it using the SMW-based algorithm instead of the SOR technique. We would like to know the combination of number of right hand sides and number of excitation frequencies that makes the SMW-based algorithm a better candidate than the SOR technique to solve the modal frequency response problem. To do so we can solve the inequality in Eq. (4.3) for the combination of number of excitation frequencies and the number of right hand sides XY. Given the modal frequency response problem with the modal structural damping matrix A_I, we found out that the combination of XY in Eq. (4.3) has to be greater than 166 for the SMW-based algorithm to perform better than the SOR technique, i.e. we can solve the modal frequency response problem for 8 right hand sides and 21 excitation frequencies faster using the SMW-based algorithm rather than using the SOR technique.

Chapter 5

Conclusions

In this chapter we discuss the outcome of the results of the DPLR approximation algorithm. First we examine the accuracy of the DPLR algorithm when it comes to representing a symmetric matrix that is known to be "diagonal plus low rank". Then we consider the DPLR approximation of general symmetric matrices. We also address the accuracy of the DPLR algorithm when a DPLR approximation is used to solve a linear system of equations. We then discuss the number of flops required to obtain a DPLR approximation with the current algorithm. We propose future work that could be done to improve the current DPLR algorithm.

5.1 Application to diagonal plus low rank matrices

When a matrix $A \in \mathbb{R}^{n\times n}$ is exactly equal to a diagonal matrix plus a low rank symmetric matrix, the DPLR approximation algorithm has proven to be very accurate in representing such a matrix. Since the function being minimized

$$\alpha = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(a_{ij} - d_i\delta_{ij} - \sum_{r=1}^{n} v_{ir}\lambda_r v_{jr}\right)^2, \tag{5.1}$$

where a_{ij}, d_i, v_{ir}, and λ_r are entries in matrices A, D, V and Λ, is continuous and differentiable for given V and Λ matrices, its minimization is relatively straightforward. The accuracy required in the approximation of matrix A may be determined by the need to solve a linear system of equations of the form $A\mathbf{x} = \mathbf{b}$. The relative error $\frac{\alpha}{||A||_F^2}$ and the gradient of the error, $\boldsymbol{\nabla}\alpha$, were used in order to determine the convergence of the DPLR algorithm to an approximation of matrix A. The accuracy of the DPLR algorithm allowed us to solve linear systems of equations in the form $(-\omega^2 I + \Omega + iA)\,\mathbf{x} = \mathbf{b}$ directly when the coefficient matrix A was exactly equal to a diagonal matrix plus a low rank symmetric matrix.

5.2 Application to general symmetric matrices

For the case in which matrix A is a general symmetric matrix, convergence of the DPLR approximation algorithm is not as simple as it was when the matrix A was exactly equal to a diagonal matrix plus a low rank symmetric matrix. In this case the best performance is achieved with an optimal choice of the rank in the low rank part of the DPLR representation. If the rank is too high, the DPLR representation is more expensive and so is each SMW-based iteration. However, if the rank is too small, more iterations are required in the solution process.

The modal FRP in Eq. (4.2) required the modal structural damping matrix A_I to be represented such that the relative error satisfied $\frac{\alpha}{||A}||_F^2 < 10^{-2}$. With this relative error it was observed that the modal FRP converged in a

few iterations. This allows us to relax the constraint that $A-(D+V\Lambda V^T)=0$ for a DPLR approximation to be considered applicable in the solution of the modal FRP.

For general matrices A, the DPLR representations are not required to be exact. Getting an exact representation of A using the DPLR algorithm can be very expensive. The fact that the representation of matrix A is not exact leads to solving systems of linear equations $[-\omega^2 I+\Omega+iA]\,\mathbf{x}=\mathbf{b}$ using the SMW-based algorithm iteratively. The number of iterations required to solve these systems of equations depend on the rank of matrices V and Λ used in the DPLR approximation of matrix A. When solving the modal frequency response problem, the fact that the entries in the matrix of natural frequencies Ω are typically about two orders of magnitude larger than the entries in the modal structural damping matrix A allows us to relax the requirement that α must be equal to zero in order to find a DPLR approximation.

5.3 Future research concerning the DPLR algorithm

Future work concerning the DPLR algorithm should be related to the choice of the initial guess of the rank of matrix V and the starting position within the V subspace. A better starting position and rank would highly benefit the DPLR algorithm in terms of the number of flops required to obtain the DPLR approximation of a symmetric matrix. It would be of great interest to explore conjugate gradient pre-conditioners in order to consistently get good convergence of the DPLR algorithm. It would be very beneficial to the

performance of the DPLR algorithm to gain a better understanding of the convergence process represented by the curve followed by the relative error measure $\frac{\alpha}{||A||_F^2}$ versus the number of iterations.

Bibliography

[1] Roger Fletcher. *Practical Methods of Optimization.* Wiley, 2nd edition, 1987.

[2] William Hagger and Hongchao Zhang. A survey of nonlinear conjugate gradient methods. Technical report, University of Florida, Gainesville, FL, February 2005.

[3] Michiel Hazewinkel. *Encyclopeida of Mathematics.* Springer, 1st edition, 2001.

[4] Martin F. Moller. A scaled conjugate gradient algorithm for fast supervised learning. Technical report, Computer Science Department, University of Aarhus, Denmark, November 1990.

[5] Elijah Polak. *Optimization: Algorithms and Consistent Applications.* Springer, 1997.

[6] Jonathan Richard Sewchuk. An introduction to the conjugate gradient method without the agoonizing pain. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburg, PA 15213, August 1994.

[7] Chang wan Kim. *Frquency Response Computation for Complex Structures with Damping and Acoustic Fluid*. PhD thesis, The University of Texas at Austin, Austin, Texas, December 2004.

Vita

David Antonio Vargas was born in Cochabamba, Bolivia on 12 June 1985, the son of Mr. Jaime D. Vargas and Rocio A. Frontanilla De Vargas. He received the Bachelor of Science degree in Aerospace Engineering from the University of Texas at Austin in May 2008, after which he was accepted to the Graduate School program at the University of Texas at Austin. He joined the Aersoapce Enginneering program in August of 2008.

Permanent address: 3100 Speedway
Austin, Texas 78758

This thesis was typeset with LaTeX[†] by 'the author'.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.